

APPLICATION FOR LETTERS PATENT

* * * * *

APPARATUS AND METHOD FOR MANAGING CALL BILLING RECORDS

* * * * *

Michael B. Freeman
Roland B. Webster
Michael D. Jarrett
Eddie L. Pickeral

DOCKET NO. COS99034

**APPARATUS AND METHOD FOR
MANAGING CALL BILLING RECORDS**

TECHNICAL FIELD

5 The present invention pertains to telecommunications data networks. More particularly, the present invention relates to the collection, format converting, and routing of call billing records via a network processor directly to a data network for billing purposes.

10 **BACKGROUND OF THE INVENTION**

 The collection of billing records from communications networks is presently carried out by switches. Individual switches within a telecommunications network, such as a PSTN network, generate billing information that is in a format which is native to the switch. In essence, the
15 switch is used to collect call records by monitoring all the dialog traffic that occurs through the switch. The billing information is then collected from all the switches within the network. However, this billing information is in a non-industry standard format which makes the collection of call billing information from multiple switches difficult and inefficient.

20 Another problem results from the need to provide expensive switch features, such as switch ports, for Internet service provider (ISP) usage. Typically, Internet service providers (ISPs) do not need expensive switch features. Likewise, Internet service providers (ISPs) often lease co-located

modems from local exchange carriers (LECs), either incumbent LECs or competitive LECs. As a result, ISPs use more floor space and spend more money on switching features and co-located modems. Therefore, there was a need to provide for new switches, many of which now present billing data in a non-
5 industry standard format. As a result, there is a further enhanced problem in that the collection of billing information from various switches is now complicated because of such non-industry standard format.

Therefore, there exists a need for improved techniques for collecting call billing information from a telecommunications network.

10

SUMMARY OF THE INVENTION

A call billing collection apparatus and method uses a local network processor as part of a switched network solution to collect billing records from a signaling gateway, convert the billing records from a raw data structure format to
15 a data format compatible with a data network, and forward the call billing data in the second data structure format to the data network for billing processing.

According to one aspect, an apparatus is provided for managing call billing records of communications network users. The apparatus includes a communications network, a gateway, a communications link, and a network
20 processor. The communications network is operative to carry user calls. The gateway communicates with the network and is operative to collect call billing data from the network in a first data structure format. The communication link is coupled to the gateway. The network processor communicates with the gateway

via the communication link. The network processor is operative to receive the collected call billing data in the first data structure format and convert the collected call billing data from the first data structure format to a second data structure format.

5 According to another aspect, an apparatus is provided for managing call billing records for users of a communications network. The apparatus includes a network, a signaling gateway, a communications link, and a network processor.

10 The network has communications capabilities to carry user calls. The signaling gateway communicates with the network and is operative to collect call billing data resulting from the calls in a first data structure format. The communication link is coupled to the signaling gateway. The network processor communicates with the signaling gateway via the communication link and is operative to convert the collected call billing data from the first data structure format to a second data structure format conducive to conducting billing processing.

15 According to yet another aspect, a method is provided for managing call billing records of users of a communications network. The method includes: providing a first computer device, a second computer device, and a communication link, the first computer device communicating with the network and the second computer device communicating with the first computer device via the
20 communication link; collecting call billing data with the first computer device in a first data structure format; transferring the call billing data using a data communications protocol from the first computer device to the second computer

device; and converting the call billing data with the second computer device from the first data structure format to a second data structure format.

According to even another aspect, a method is provided for managing call billing records generated from usage within a communications network by users.

- 5 The method includes: providing a signaling gateway communicating with the network and a network processor communicating with the signaling gateway; collecting call billing data with the signaling gateway in a first data structure format; transferring the call billing data using a data communications protocol from the signaling gateway to the network processor; and converting the call
- 10 billing data with the network processor from the first data structure format to a second data structure format conducive to processing billing information.

One advantage is the ability to provide billing records to a data network in an industry standard format.

- Another advantage is the compatibility between the network processor and
- 15 existing local traffic systems.

BRIEF DESCRIPTION OF THE DRAWINGS

- Fig. 1 is a schematic block diagram illustrating the architecture of a telecommunications network capable of benefiting from the apparatus for
- 20 managing call billing records of Applicant's invention which is described in greater detail with reference to Figures 2-3.

Fig. 3 is a logic flow diagram illustrating the steps involved in

5

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention is directed towards a telecommunications call collection apparatus that is able to present billing record data to a data network in a format conducive to processing of billing information. Accordingly, billing records are retrieved from a communications network and converted into another format that is conducive to such processing. According to one implementation, the converted format is a recognized industry standard data format.

Figure 1 is a schematic block diagram depicting one embodiment of the present invention in the form of a call billing collection apparatus identified by reference numeral 10. Apparatus 10 communicates with a telecommunications network 12 comprising a public switched telephone network (PSTN) 14, one or more subscribers 16, the Internet 18, Internet service providers 20, an integrated access platform 22, a network processor, or platform, 24, and a data network 26.

According to one implementation, PSTN network 14 is a PSTN signaling system 7 (SS7) network 28. SS7 network 28 uses a signaling system referred to as an ITU signaling system 7 (SS7) which is a newer out-of-band signaling system that is being required by many telecommunications administrations worldwide for their networks. ITU (in French) refers to the International Telegraph and Telephone Consultative Committee that approved the ITU SS7 recommendations that have been optimized for digital networks. The resulting protocol uses destination routing, octet oriented fields, variable length messages and a maximum message length allowable for 256 bytes of data.

As shown in Figure 1, signaling gateway 30 is an Ascend signaling gateway manufactured by Ascend Communications, Inc., of Westford, Massachusetts, recently merged with Lucent Technologies, Inc., of Murray Hill, New Jersey. Integrated access platform 22 provides call billings record management apparatus
5 10 and a network access server (TNT) 32, also manufactured by Lucent Technologies, Inc. Apparatus 10 includes signaling gateway 30 and network processor 24.

PSTN SS7 network 28 includes one or more signal transfer points (STPs), such as an incumbent local exchange carrier (ILEC) STP 34 and a network
10 telecommunications services provider STP 36 that is also providing the call billing services of apparatus 10.

Network processor, 24 is configured to provide call collection functionality according to one aspect of Applicant's invention. More particularly, network processor 24 is configured to poll integrated access platform (IAP) 22
15 every 15 minutes in order to collect call billing records from network 12, in this case, PSTN 14. Network processor 24 then converts the collected call billing records into a Bellcore automatic message accounting (AMA) format, wherein AMA provides an industry-recognized standard for billing information. Network processor 24 then sends the AMA records and audit reports to a data network 26,
20 such as a local traffic system (LTS) 42. LTS 42 then mediates the AMA records, and forwards such records to a corporate system data feed (CSDF) 44. Subsequently, CSDF 44 forwards the AMA records to one or more co-carrier access billing systems (CCABS) 46. CCABS 46 then places the resulting records

into its database for settlement with each incumbent local exchange carrier 33 and each internet service provider (ISP) 20.

Accordingly, NP 24 of IAP 22 is configured to collect, convert, and deliver call billing records to LTS 42. NP 24 is provided with an interface 48 that
5 connects with IAP SS7 gateway 30 and enables polling, at preset intervals, to collect raw ASG call event records (CERs) 50. NP 24 is operative to invoke a reformatting process that converts the ASG CERs 50 into a Bellcore AMA format (BAF) 52. BAF 52 records are written and stored at NP 24, then sent to LTS 42 for further billing processing.

10 ASG CERs 50 provide collected call billing data in a first data structure format 54. Network processor 24 then converts such collected call billing data from first data structure format 54 into BAF 52 which is in a second data structure format 56 that is recognizable by the network processor and readily usable by a data network 26.

15 Network processor 24 is provided with the following features. First, call billing file collection is carried out from ASG 30 via a data communications protocol. Call billing data is transferred from gateway 30 to network processor 24 using the data communications protocol from gateway 30 to network processor 24. According to one construction, gateway 30 is provided by a first computer
20 device, and network processor 24 is provided by a second computer device. According to one implementation, the data communications protocol is a file transfer protocol (FTP).

Secondly, billing record conversion is carried out from an ASG native format to a Bellcore AMA Structure Code 625 format. Thirdly, call billing is delivered to LTS 42 using the file transfer protocol (FTP). Fourth, network processor 24 implements an audit report that uses a reconciliation technique that tracks the summation of files that are sent to a data network, or a billing system.

Network processor 24 sends such files to data network 26. The reconciliation technique comprises a software routine wherein the number of bytes of data is counted, then the counted-up number of records and/or bytes of data are compared to a status file in order to determine when a full, successful transfer has occurred.

Furthermore, network processor 24 provides WEB-based audit tool reporting features, and one or more alarm signal reporting to a Tivoli alarm management system, sold by Tivoli Systems, Inc., of Austin, Texas.

According to one implementation, data network 26 is a local traffic system (LTS).

Several benefits are provided by call billings record management apparatus 10, via network processor 24, as follows. First, ASG billing records from SS7 network 28 and network access server 32 are converted into an industry standard AMA format. Second, network processor 24 provides connectivity and billing record data flow capabilities into an existing local traffic system (LTS) 42 presently available from MCI Worldcom having an MCI Worldcom data network frame relay/ATM (not shown). Third, network processor 24 delivers billing traffic to co-carrier access billing systems (CCABs) 46 for settlement. Fourth, network processor 24 is an in-house developed application that is compatible with

the MCI Worldcom LTS 42 which facilitates rapid development and deployment within existing MCI Worldcom network environments.

Figure 2 illustrates an integrated access platform (IAP) network processor (NP) call collection architecture for use with call billings record management apparatus 10 of Figure 1. More particularly, call processing is carried out between signaling gateway 30 and network processor 24 by implementing call processing at network processor 24. In order to carry out such call processing, the following steps are implemented via network processor 24.

First, network processor 24 polls the raw ASG CERs 50. Then network processor 24 constructs a BAF output file. Subsequently, network processor 24 pushes the BAF data files to LTS 42. Retransmission is then carried out along with re-polling, followed by an archiving/restoration operation. Finally, a call record view is provided for CER data 50 and BAF data 52 (see Fig. 1).

In order for network processor 24 to poll raw ASG CERs 50, network processor 24 implements the following steps. First, network processor 24 logs in to ASG 30. Subsequently, network processor 24 goes to a call event records (CER) directory. Then network processor 24 transfers the file as text and/or binary files from gateway 30 to network processor 24 at predetermined intervals using a file transfer protocol (FTP). Such transfer, according to one implementation, is carried out at a default setting for the predetermined intervals of 15 minutes. Next, verification is made by network processor 24 that the file transfer has been successful. Subsequently, the ASG file is renamed by the

network processor 24. Finally, network processor 24 moves the file to a backup directory.

In order to construct an output file containing the Bellcore AMA format (BAF) data 52 (see Fig. 1), network processor 24 implements several steps.

- 5 Network processor 24 first converts each file containing Lucent CER data 50 to a file containing BAF data 52 in a BAF 625 format. Additionally, network processor 24 will populate the listed BAF 625 data elements with the following default values:

- | | | |
|----|--------------------------------|--|
| 10 | (1) Sensor Type = 26 | |
| | (2) Sensor ID = Initially 1, | for first ASG and
increments by one for
each additional ASG |
| | (3) Recording Office Type = 26 | |
| 15 | (4) Recording Office ID = | Initially 1, for first ASG and
increments by one for each
additional ASG |

- Network processor 24 then prepends the output file containing BAF data 52 with a BAF 9038, logical data set header record which is a specific billing format
- 20 defined by the Bellcore Automatic Message Accounting (AMA) Format (BAF) Requirement TR#030#NWT-001100. Such format identifies paths according to the resource they are terminated on. Additionally, network processor 24 will populate the BAF 9038 data elements with default values as follows:

- | | |
|----|-----------------------------|
| 25 | (1) AMA Sequence Number = 0 |
| | (2) Program Generic No. = 0 |
| | (3) Type of Tracer = 28 |
| | (4) Header Type = 0 |

- (5) Sending Unit Number = 0
- (6) First Block Sequence Number

Furthermore, network processor 24 postpends the BAF output file with a
5 BAF 9039, logical data set trailer record. Additionally, network processor 24 will
populate the BAF 9039 data elements with default values as follows:

- (1) DCC/APS Program Generic No. = 6
- (2) Type of Tracer = 29
- (3) Trailer Type = 0
- 10 (4) Last Block Sequence Number
- (5) Block Count = 3

In order to construct the BAF output file, network processor 24 will provide
block sequence numbers that will be simulated for the integrated axis platform call
15 collection application of apparatus 10. Network processor 24 will provide a
gateway specific running counter of block sequence numbers for each signaling
gateway 30 (of Fig. 1) that is polled. The block sequence number will range
between 00,000,001 to 99,999,999, where 00000000 is reserved for system
restarts. Network processor 24 will increment the block sequence numbers for
20 each block that is created. When the first block's sequence number is greater than
the last block's sequence number, the first block shall have a sequence number
equal to 1, and the subsequent block numbers will be incremented accordingly.

Such event will occur after there has been a system restart, and the network
processor 24 reaches 99,999,999 for a block sequence number for any block other
25 than the last block in the file.

Each file that is sent to local traffic system (LTS) 42 by network processor 24 will have the equivalent of three blocks (see Fig. 1). The three blocks are defined as follows:

- (1) Logical Data Set Header (Structure Code 9038) record
- 5 (2) All the Terminating Access (Structure Code 0625) records
- (3) Logical Data Set Trailer (Structure Code 9039) record

In order for processor 24 to push IAP BAFs to local traffic system (LTS) 42 (of Fig. 2), the following steps are required. First, network processor 24 logs
10 into LTS 42. Subsequently, network processor 24 uses the default directory. Next, network processor 24 transfers the BAF output file via file transfer protocol (FTP) to local traffic system (LTS) 42 with a temporary file name. Subsequently, network processor 24 verifies that the file transfer has been implemented successfully. Following successful verification, network processor 24 transfers
15 the reconciliation file via file transfer protocol (FTP). Next, network processor 24 verifies that the reconciliation file was received correctly by LTS 42. Finally, network processor 24 renames the BAF output file following the LTS naming conventions.

In order for network processor 24 to provide re-transmission and re-polling,
20 the re-transmission and re-polling is manually initiated via a standard network processor (NP) user interface provided for on-line files. When the files are not on-line, a manually initiated file transfer will be utilized. First, network processor 24 will provide re-polls from on-line ASG files. After performing a re-poll, network processor 24 will convert the CERs to BAFs and send them to local traffic

system (LTS) 42 using the same file names and sequence numbers as the original file. Subsequently, network processor 24 will perform re-transmission to local traffic system (LTS) 42 from locally on-line BAF output files.

In order for network processor 24 to archive and/or restore CER and BAF files, network processor 24 will provide CER and BAF archive capabilities to an Adstar Distributed Storage Management (ADSM) subsystem, and restoration from the ADSM subsystem.

In order for network processor 24 to provide a call record view, network processor 24 provides the call record view for both CERs and BAFs. The user will input a file name in order to open the file. The user will then be able to scroll through the file and view the call records in a user-friendly manner. In order to realize such implementation, a "call search" will be provided therein.

Several interface requirements exist where network processor 24 interfaces with gateway 30 and LTS 42. An interface between gateway 30 and network processor 24 is achieved as follows. First, the ASG CER files containing raw CER data 50 (of Fig. 1) need to be read by network processor 24. Secondly, each read and processed CER file is renamed. Thirdly, the renamed CER files are moved from one directory to a new directory.

With respect to the first step above, network processor 24 will read CER files from signaling gateway 30, here an Ascend Signaling Gateway (ASG), then rename the CER file after processing it. Subsequently, network processor 24 moves the CER file to a backup directory.

In order for network processor 24 to read the ASG CER files, network processor 24 will connect to signaling gateway, Ascend Signaling Gateway (ASG) 30 using internet protocol (IP). Subsequently, the ASG CERs will be file transferred, via file transfer protocol (FTP), from an ASG directory. One suitable
 5 ASG directory is *"/home/cfgmgr/log/cer."*

For the case where multiple ASG multi-stack nodes are running on a given ASG platform, the CER files are located in a specific "cfgmgr", and one exemplary directory location is *"/home/cfgmgrN/log/cer"*, where N specifies a specific instance of the ASG multi-stack node. Within each instance of the ASG
 10 multi-stack node, there may be multiple files based on the instances of modem call managers (MCMGRs).

The ASG CER file naming will be mcmgrn-yyyymmddHHMM.dat (file names are case sensitive) where:

	mcmgr	are the fixed characters "mcmgr"
15	n	is the instance of the Modem Call Manager (range 1-5)
	-	is the fixed character "-"
	yyyy	is the file creation year
	mm	is the file creation month
20	dd	is the file creation day of the month
	HH	is the file creation hour, based on 24-hour clock
	MM	is the file creation minute
	.dat	are the fixed characters ".dat"

25 In order to rename process CER files, network processor 24 will rename the CER file on the ASG after verifying that the file transfer has been successful. The

file renaming convention will change the file suffix from ".dat" to ".old". However, the remainder of the file name will remain the same. For purposes of uniformity, a renamed file will have the following syntax:

mcmgrn-yyyymmddhhmm.old

5 In order to move renamed CER files, network processor 24 will move the renamed CER file on the ASG from its current directory into the following subdirectory "/home/cfgmgr/log/cer/cer_bak". For this case, the file and the source directory will be deleted. The subdirectory "cer_bak" will have been created during system configuration.

10 An interface 66 between network processor 24 and LTS 42 is achieved as follows. First, network processor 24 pushes the BAF output file, containing BAF data 52 to LTS 42 (of Fig. 1). Secondly, network processor 24 sends a reconciliation file after successfully sending each BAF output file to LTS.

15 At the interface between network processor 24 and LTS 42, network processor 24 will push the BAF output file to local traffic system (LTS) 42 per an ASG poll. For the case where multiple ASGs are implemented, multiple BAF output files will be sent to local traffic system (LTS) 42. According to such implementation, the BAF output file structure will be as follows:

	<u>Structure Code #</u>	<u>Definition</u>
20	9038	Logical Data Set Header
	625	Terminating Access Records
	<u>9039</u>	<u>Logical Data Set Trailer</u>

The BAF output file naming convention will be
"miap.nnnn.ccyymmdd.HHMMSS.swid", with file names being case sensitive,
 where:

	miap	are the fixed characters "miap"
5	.	is the delimiter character "."
	nnnn	is the file sequence number assigned by NP having a range 0001-9999. 0000 is reserved for system restarts. The NP will provide an ASG specific running file sequence number for each unique ASG or instance of an ASG.
10	.	is the delimiter character "."
	ccyy	is the file creation year
	mm	is the file creation month
	dd	is the file creation day of the month
15	HH	is the file creation hour, based on 24-hour clock
	MM	is the file creation minute
	SS	is the file creation second
	.	is the delimiter character "."
20	swid	is a five-character ID assigned to each ASG using the syntax ccnn where cc is a two-character city abbreviation and nn is a two-digit number having a range from 01 to 99. Example switch IDs are irg01, da102, and nyk53.

25 It is important to note two things. First, used values are extracted from the
 ASG CER file name. Secondly, seconds are computed from the date/time of the
 answer field of the first BAF in the file.

The temporary file naming conventions will prepend **"TEMP."** to the BAF
 output file name until the file transfer is successfully validated, and the
 30 reconciliation file transfer is validated.

nnnn is the file sequence number assigned by NP
 having a range 0001-9999. 0000 is reserved for
 system restarts. The NP will provide an ASG
 specific running file sequence number for each
 unique ASG or instance of an ASG.
 . is the delimiter character “.”
 ccyy is the file creation year
 mm is the file creation month
 dd is the file creation day of the month
 mm is the file creation minute
 ss is the file creation second
 . is the delimiter character “.”
 swid is a five-character ID assigned to each ASG using
 the syntax ccnn where ccc is a three-character city
 abbreviation and nn is a two-digit number having
 a range from 01 to 99. Example switch IDs are irg01,
 da102, and nyk53.

It is important to note two things. First, used values are extracted from the
 ASG CER file name. Secondly, seconds are computed from the date/time of the
 answer field of the first BAF in the file.

The temporary file naming conventions will prepend “TEMP.” to the BAF
 output file name until the file transfer is successfully validated, and the
 reconciliation file transfer is validated.

With respect to the integrated access platform (IAP) reconciliation file,
 network processor 24 will send a reconciliation file after each BAF output file is
 successfully sent to local traffic system (LTS) 42. The reconciliation file name
 according to one implementation is “*miap.nnnn.ccyyymmddhhmmss.swid.SR.dat*”.

With respect to the integrated access platform (IAP) reconciliation file, network processor 24 will send a reconciliation file after each BAF output file is successfully sent to local traffic system (LTS) 42. The reconciliation file name according to one implementation is "*miap.nnnn.ccyymmddhhmmss.swid.SR.dat*",

5 using the same time stamp and sequence number as the BAF output file. The reconciliation file will contain the following data elements:

Field	Format	Comments & Examples
Switch ID	Up to 6 alphanumeric Characters	IRG01, ATLA, BALT22, LOSANG, etc.
First block sequence number of the Logical Data Set (LDS) Block	Integer	Taken from the 9038 record
Date that First Block was Written to Disk	Char CCYYMMDD	Taken from the 9038 record
Time that First Block was Written to Disk	Char HHMMSS	Taken from the 9038 record
Last block sequence number of the LDS Block	Integer	Taken from the 9039 record.
Date that Last Block was Written to Disk	Char CCYYMMDD	Taken from the 9039 record.
Time that Last Block was Written to Disk	Char HHMMSS	Taken from the 9039 record.
Number of Records excluding Headers and Trailers	integer	The number of 625 records in the file.
Number of Blocks	integer	The MIAP NP file contains 3 blocks.

As shown in Figure 2, alarm signals 24 are delivered from network processor 24 to a Tivoli alarm management system. Network processor 24 has multiple levels of alarm

10 classifications. According to one implementation, the range of alarm classifications will be:

a) informational, b) minor, c) major, and d) critical. An additional or optional feature

consists of enabling the emailing of alarms 24 to additional monitoring systems (not shown).

Network processor 24 will generate the following list of alarms, and will deliver them to a Tivoli alarm management system (not shown), such as is provided within several Tivoli Management packages presently sold by Tivoli Systems, Inc., of Austin, Texas:

- 5 1. Duplicate records based on CER sequence number.
2. Missing records based on CER sequence number.
3. Out of Sequence records based on CER sequence number.
4. No data received from any generating system in 6 hours.
5. Port failures.
- 10 6. Polling failures by any sending unit.
7. Mass storage/write errors.
8. Program errors.
9. Hardware failures.
10. Archive tape failures.
- 15 11. Power off.
12. Mass storage of: 70%, 90%, 100% (default), or "user-set" % for full.
13. Invalid password.
14. File interrupt at sending unit.
15. Time out on data links.
- 20 16. Poll queued to invalid port.

Network processor 24 of Figure 2 is configured to produce reports at preset intervals and on-demand. The user will have options to request that the reports are displayed on a video screen, output to a printer, or emailed. Following are exemplary reports:

1. Daily volume tracking inputs and outputs by structure code, call code, IXC ID code.
2. Hourly volume tracking inputs and outputs by structure code, call code, IXC ID code.
3. Hour-by-hour tracking average based on three previous days' data.
4. Polling schedule report listing session activity for receiving CREs and sending BAFs:
 - A) file transfer session start time, initial record sequence number;
 - B) file transfer session duration, final record sequence number;
 - C) bytes received or sent; and
 - D) session status, i.e. successful, aborted, not data available, target system not responding.
5. Alarm report listing the last 1000 alarms by date, time, classification, and description.
6. System performance report over a previous five day window, by day, listing:
 - A) records received, records sent;
 - B) CPU utilization, disk utilization, communication utilization; and
 - C) successful sessions, failed sessions.

System requirements for carrying out one exemplary implementation of the call billings record management system of Applicant's invention entail the following storage requirements, network communications requirements, and processing capabilities. Such exemplary system requirements have been calculated based on the assumption of one Ascend (Lucent) signaling gateway (ASG), configured with 100,000 ports, and rated at 270,000 busy hour call attempts.

For one set of exemplary storage requirements, the network processor 24 (of Fig. 2) will be assumed to retain the CERs and BAFs on-line for 31 days. The resulting storage requirements, for one ASG, are computed as follows:

$$\begin{array}{l} \text{ASG CER} = 92 \text{ bytes} \\ \text{BAF } 625 = 83 \text{ bytes} \\ \text{Total bytes per call} = 175 \text{ bytes} \end{array}$$

Each ASG can process 270K calls per busy hour. A busy hour is typically 10% of a day's traffic. Therefore, there can be 2.7 million calls per day. The resulting total bytes per day can be calculated, as a result, to be:

$$175 \text{ bytes/call} * 2.7 \text{ million calls/day} = 472.5 \text{ million bytes.}$$

Total storage for 31 days can then be calculated as:

$$472.5 \text{ million bytes/day} * 31 \text{ days} = 14.65 \text{ gigabytes.}$$

Network communications requirements can also be calculated by example. The network communications bandwidth, for one ASG, is calculated based on data transfer during a busy hour. This number is then doubled to account for the busy second using the following algorithm:

$$\begin{array}{l} \text{Record size (in bits)} * \# \text{ of records in busy hour} / 3600 \text{ seconds/hour} * 2 \\ \text{(busy second factor)} = \text{bits/second} + 10\% \text{ for retransmission and miscellaneous} \\ \text{overhead.} \\ \text{Input} = 121 \text{ kb/s} \\ \text{Output} = 109.5 \text{ kb/s} \end{array}$$

Processing capacity requirements for network processor 24 will be sufficiently sized to complete one polling cycle before beginning the next polling cycle. The polling cycle is defined as:

1. Input CERs, convert CERs to BAF, and output the BAF output file.
2. Process and output the IAP reconciliation file, rename the BAF output file.
- 5 3. Statistic and peg count accumulation for reporting.
4. Alarm reporting.

10 Figure 3 discloses a first level logic flow diagram for the programming of processing circuitry present within apparatus 10, and more specifically, within gateway 30 and network processor 24 (of Fig 1). The logic flow diagram describes a process that includes logical operations that are implemented by the processing circuitry within the call billings record management apparatus.

15 Furthermore, the logical flow diagram includes steps utilized according to one implementation of Applicants' invention.

In Step "S1", signaling gateway and a network processor are provided in communication therebetween. After performing Step "S1", the process proceeds to Step "S2".

20 In Step "S2", the process collects call billing data with the signaling gateway in a first data structure format. After performing Step "S2", the process proceeds to Step "S3".

In Step "S3", the process transfers the call billing data using a data communications protocol from the signaling gateway to the network processor

25 After performing Step "S3", the process proceeds to Step "S4".

In Step "S4", the process converts the call billing data with the network processor from the first data structure format to a second data structure format that

In Step “S5”, the process routes call billing data, or data records, for a user via the network processor to a data network. It is understood that Step “S5” is an optional step. After performing Step “S5”, the process terminates

23-24